Is Neural Network Verification Useful and What Is Next?

Taylor T. Johnson
Institute for Software Integrated Systems
Vanderbilt University
Nashville, TN, USA
taylor.johnson@vanderbilt.edu

***Abstract*—Neural network verification is an instantiation of the classical formal verification problem, which is, given a model of a system and a specification, both with precise syntax and semantics, prove that the model satisfies the specification. The neural network verification problem instantiates this generic problem, where the model is a neural network represented as a mathematical function and the specification is typically defined as subsets over the input and output spaces of the function, where the model satisfies the specification if all inputs map only to desired outputs. The past decade or so has witnessed significant progress on this problem, such as improved scalability (e.g., in the sizes of the neural network and specifications), application on a variety of machine learning tasks, and important case studies in critical domains ranging from autonomy to medicine, among other advancements. However, is this progress so far useful, what are the current strengths and weaknesses of results in this research area, and what's next for this research area and related problems? This position presentation will take a critical view of this research area discussing these aspects, highlighting where it is and is not useful, and present suggestions for what problems to investigate next.**

***Index Terms*—neural network verification, safe AI, neural networks, machine learning, artificial intelligence, formal methods**

I. Introduction

Neural network verification [1]–[41] is an instantiation of the classical formal verification problem, and is: given a (trained) neural network and a specification, verify the network satisfies the specification. The typical formulation considers a neural network represented as a function $f : X \mapsto Y$ when given a symbolic subset of inputs $I \subseteq X$ computes the image of f under I, that is $f(I)$, and shows this is contained in a symbolic output set $O \subseteq Y$, that is $f(I) \subseteq O$. Typically, a real abstraction is used, so $X = \mathbb{R}^n$ and $Y = \mathbb{R}^m$, but there are considerations of numerical representation (e.g., floating-point, fixed-point, etc.) or quantization that may influence results due to numerical precision. Many formulations and approaches of the problem exist, ranging from abstract interpretation, reachability, and interval bound propagation (IBP) to constraint-based formulations of optimization problems often as mixed-integer linear programs (MILPs) or satisfiability modulo theories (SMT).

Over the past decade especially, many research projects, results, papers, research community events, conference and journal special sessions, competitions, and software tools addressing this problem have emerged. For example, in the International Verification of Neural Networks Competition (VNN-COMP) held annually since 2020, several dozen verification tools have participated, including α, β-CROWN [9], [37], [42], [43], AVeriNN, CGDTest [44], CORA [31], Debona [45], DNNF [46], DNNV [47], ERAN (encompassing DeepZ [12], DeepPoly [13], AI2 [7], etc.), FastBATLLNN [48], Marabou [15], MIPVerify [11], MN-BaB [49], NeVer2 [50], NeuralSAT [51], nnenum [52], NNV [8], [16], [19], [53], ModelVerification.jl/NV.jl [54], OVAL [55], PeregriNN [56], PyRAT [57], Reluplex [1], RPM [58], VeRAPAk [59], Venus [60], VeriNet [61], among others, on dozens of benchmarks, several from industry [62] and from diverse domains ranging from medical imaging [63], aerospace [64]–[66], autonomy [67], and security [68]. As another example, there is significant work considering verification where neural networks are used as feedback controllers for dynamical systems (plant models), such as in the hybrid systems verification (ARCH-COMP) Artificial Intelligence and Neural Network Control Systems (AINNCS) category [69]–[74], with tools such as COntinuous Reachability Analyzer (CORA) [75], [76], CROWN-Reach, immrax [77]–[79], JuliaReach [80], NNV [19], [53], Sherlock [4], [81], [82], POLAR [83], OVERT [84], ReachNN* [85], [86], VenMAS [87], and Verisig [88], [89].

In spite of these significant efforts and results (and many others [90]–[94]), this paper presents a critical evaluation of the status of neural network verification, where it is useful (and not), and directions for future research in the development of formal methods for establishing safe, secure, and trustworthy AI.

II. Is Neural Network Verification Useful?

Is neural network verification useful? Yes and no. While the neural network verification problem is attractive due to its elegant statement, there are critical gaps in most works to make it useful (including the author's). Verifying a neural network alone is not particularly useful: they are not used alone. Of course, aspects of neural network verification are useful in this context, in that the neural network can in essence be replaced in the broader system with an assume-guarantee contract, in essence, rewriting the call to the neural network with a pre-post condition. However, for most use cases where neural networks

make sense, it is not clear if this makes sense due to a key aspect, which is data. While this is a critical task, it is essential to integrate the results of verification into broader system workflows. Some work exists in these directions, namely in autonomy, such as with verification of neural network control systems, where a neural network is utilized as a feedback controller for some plant model, often modeled as ordinary differential equations (ODEs) [95]. However, there are several missing points, which exist for several reasons.

a) Problem 1: Neural Networks are Not Used Alone: First, it is challenging to precisely specify problems around integration of neural networks used within broader systems. If a computation utilizes the result of a neural network call, we would ideally like to verify the result of this overall computation. Some neuro-symbolic programming languages (such as Scallop [96], neuro-symbolic behavior trees [NSBTs] [97], [98], neuro-symbolic automata [99], and others [100], [101]) partly address this, but often keep these calls as holes without guarantees. This perhaps partly is due to a lack of integration between the programming languages, formal methods, AI, and machine learning communities, albeit there are some works in this direction as illustrated above. However, there has yet to be a strong integration between the neural network verification problem and these neuro-symbolic approaches, and this is a clear opportunity for future research, which also addresses this problem in most existing works and one the community should focus toward [102].

b) Problem 2: Certified Robust Accuracy (CRA) May Not be Useful: In a large number of papers, neural network verification methods are evaluated primarily on image classification tasks, namely to quantify what is generally called certified robust accuracy (CRA). First, the neural network verification problem is instantiated for an image classification problem to compute local adversarial robustness up-to ϵ with respect to a data point $x \in X$. Then for a fixed ϵ, an L_p ball is defined up-to ϵ away from the data point $x \in X$, and the neural network is locally adversarially robust up-to ϵ about x if all points in the L_p ball are also classified the same. Most commonly the L_∞ norm is used, although there are issues with this, and a variety of works have considered other distance measures. CRA is then computed as the average across all test points that are robust. Intuitively, as ϵ increases, CRA decreases, as at some point decision boundaries are crossed.

CRA in essence gives a metric similar to accuracy, but for indicating how robust the neural network is under perturbations. Computing CRA of course is useful for evaluating scalability of the verification methods in the size of the neural network and the size of the specification (as it gets harder as ϵ increases), it is not clear if this application makes any sense beyond an easy way to parameterize the verification problem to evaluate and demonstrate scalability of the verification methods. Is a neural network with higher CRA better than a neural network with lower CRA? If so why? Our suggestion is to stop looking at CRA, unless it can be demonstrated that quantifying CRA provides usefulness beyond evaluating scalability of the verification methods easily, and instead, the community should focus on verifying specifications that make more sense.

c) Problem 3: What are the Specifications?: The grand challenge problem in the neural network verification domain is what are the specifications that should be verified? As the discussion on CRA above illustrates, we need to develop, illustrate, and evaluate the benefits of neural network verification by clearly illustrating how it can be used to verify useful specifications. There are some instances of this, but more case studies, frameworks, etc., are needed. Most likely, these must be developed in task-specific and application-domain specific contexts, as is common in the development of formal specifications in different areas. We strongly encourage the community to support the development of this direction, as the specification problem is the grand challenge: how do we say when AI-based systems are correct? Of course this is related to other ongoing challenges more broadly, and likely is related if not equivalent to the alignment problem in AI.

d) Problem 4: Neural Networks are not Really Just Mathematical Functions: While the mathematical abstraction that a neural network is just a function is elegant and makes the problem precise, the implementations of neural networks are not simply functions, and instead, rely on sophisticated implementation in tensor-based computations. Most if not all works ignore this detail, which is important in practice, as has been illustrated with some papers demonstrating soundness problems in neural network verifiers. For example, the details of underlying architecture where the computations are performed are important, as has been demonstrated by several results around the soundness or lack thereof in many neural network verification approaches [103]–[108]. The underlying computation architecture (e.g., ISA) that perform the computations is crucial, and more broadly, motivates the formalization of pieces of this problem going beyond existing efforts in VNN-LIB [109] and ONNX. A formal semantics of ONNX and other interchange formats is needed, as are those going down to the architectural level (e.g., at CUDA or intermediate representation like LLVM).

e) Problem 5: How Does Neural Network Verification Help Create Better Neural Networks?: While there are several works around neural network repair and robust training [110]–[113], it is not clear if neural network verification aids in developing better neural networks. While CRA evaluations demonstrate some form of robustness, it is well known simply due to the existence of decision boundaries for classification problems, that global robustness is impossible: there will always be points near the decision boundary that slightly perturbed cross it. A strong argument however could exist, for instance, if neural networks that are robust as measured via high CRA may provide better generalization capability. However, this type of evaluation has not yet been fully demonstrated, and is another opportunity to demonstrate the potential usefulness of neural network verification.

f) Other Problems: Aside from these issues, there are many other aspects not significantly considered in neural network verification that limit its usefulness. What is the role

of datasets in neural network verification? If a task is complex enough to rely on data-driven approach and has a substantial dataset, can we ever consider verifying all or even a reasonable subset of the possible valid data? If we don't, have we verified anything? The limitation to local analysis, and the significant challenges in addressing global analysis, vastly limit current usefulness of neural network verification.

III. What's Next?

Next, I outline several research directions within and beyond neural network verification.

a) Integration with Workflows: A key research need is the integration of neural network verification within broader programming languages, frameworks, and systems.

b) Semantics: Foundational, Interchange, and Compilation: While the evolution of machine learning frameworks (e.g., from Matlab to Caffe to TensorFlow to Keras to Pytorch to ONNX to Jax to whatever is next) have substantially impacted the progress of machine learning, there are significant gaps in the semantics of neural networks. This of course is crucial for neural network verification. Some progress on this front has been made in the community with some standardization efforts (e.g., the usage of ONNX and VNN-LIB), the semantics of these are not fully defined. Further, even if we suppose the semantics of ONNX are precisely defined (albeit this is unrealistic as there are many versions), there is another crucial issue that would be ignored: it would likely not account for the underlying hardware implementation details (e.g., to CUDA or Trainium or TPUs or whatever else), which is crucial for the results to hold on the hardware that will execute the model. Some effort has gone into consideration of some numerical issues, as well as implementation-specific details (e.g., quantization in embedded domains). Due to the trend toward all major cloud vendors developing in-house accelerators for rather obvious economics, this issue will become even more crucial over the next, as differences in implementation (e.g., compilation) to the target hardware on which the neural networks will execute may yield soundness problems. However, significantly more work in this direction is needed, as for example, several recent works have pointed out theoretical and practical soundness issues with most if not all existing neural network verification methods (both due to theoretical issues as well as implementation issues around numerics in floating/fixed point, etc.).

c) Generative AI: What is the role of neural network verification in generative AI, and especially language models? Of course while one can formulate computing say CRA or a similar metric for a large language model (LLM) like ChatGPT (albeit an open weight model like Llama [114] or more realistically OLMo [115], [116] and even more realistically some small language model [SLM] for scalability), it is not clear what this would mean, albeit there are some results in this direction of considering natural language processing (NLP) tasks and the crucial embedding gap [117], [118], along with considerations such as the tokenization process. However, considering how to use neural network verification in generative AI leads to an important thought experiment and challenge detailed below. Other generative tasks may be more precise to specify something more meaningful, but it is still not clear and so more research is needed in this direction. Further, there would be many challenges beyond this specification issue, to scalability and architectural support (layer types, etc.), although from recent VNN-COMP results [26], we may now be within the real of possibility to verify something for SLMs with on the order of a billion parameters. While there are some papers on attention layer verification as needed in transformer architectures [119]–[121], significantly more work in this direction is needed to ensure all parts of a language model architecture could be supported for neural network verification.

All this said, due to business cases and economic considerations, it is likely industry will devote heavily in making effective smaller language models, which may naturally mitigate the scalability concerns that make this idea at first glance sound ridiculous. As argued in some recent papers, due to latency needs in agentic AI, as well as cost considerations (e.g., token cost), it is likely the AI industry will focus heavily on decreasing the size of models to reduce latency and cost [122]. Given this, the first glance concern that neural network verification cannot scale to language models may naturally be mitigated, as the models will become smaller over time with good performance.

d) Applications: While neural network verification has been applied to many problems and case studies in numerous domains (autonomy, medicine, security, etc.), some killer applications of the problem are still missing, especially where the verification has been shown to be impactful in real systems. The research community can engage with industry, and also industry can engage with the academic research community, to develop meaningful neural network verification benchmarks that have specifications of importance and interest for real systems.

e) Specifications: The input-output specifications considered in most works are inadequate. New specification languages, both for neural networks and their usage in broader systems, are needed. For example, suppose within a larger system a neural network is executed several times, with subsequent computations dependent upon prior calls, but in combination with postprocessing of the results of the neural network computations. What program logics could specify such behaviors? Further and more fundamentally, especially for perception tasks where neural networks are the best known approach in many such computer vision problems, the overall integration of neural network verification and program/software verification must be explored further.

f) Call to Action: Verify a Language Model and Agentic AI System: For practical reasons, verifying an LLM like ChatGPT is impossible for the research community, as much of it is proprietary. In addition to scalability concerns, there are dataset concerns that would limit what a meaningful specification could be, along with the challenge that the neural network must be available to perform neural network

verification. However, there are now some fully open language models, covering datasets through code and weights, such as Allen AI OLMo2 [115], [116], where the smallest model is OLMo2 1B. Based on the challenges and opportunities identified in this paper, we suggest the research community to seriously considered what it would take to verify some types of specifications for an LLM (or SLM)? Initial robustness-like specifications are easy to formulate (as they are similar to local adversarial robustness), but these should be expanded. For NLP tasks, this could amount to Hamming or Levenshtein distance perturbations or generalizations thereof, but should expand to be more meaningful for task specific purposes. Once all parts of an SLM can be considered with the layer-type support, architectural consideration, and reasonable scalability, attempting to verify an SLM as used in agentic AI system with further processing and program logic is an obvious next step. Altogether, the directive is to verify some specifications of an SLM, akin to the impactful challenge of Hoare to develop a verified compiler [123], [124], which motivated significant progress and several significant results like CompCert [125] and VST [126]. Verifying an SLM is a similar directive and challenge for the community, which will naturally lead to progress on all these challenges and limitations in neural network verification currently, and even if failing, will significantly advance the status and impact of neural network verification.

IV. Conclusions

In summary, neural network verification has not yet demonstrated its full usefulness and potential impact. While it is useful for some domains and certain restricted tasks, there are many critical issues in the research area that need to be addressed, as well as potential avenues forward to help address those limitations, so that the research area may truly make an impact. As detailed some above, we suggest a challenge to try to verify a small language model (SLM) in the context of an agentic AI system, which likely will require a neuro-symbolic verification approach for a neuro-symbolic system, combining neural network verification with software and program verification. If nothing else, verifying a language model is a useful thought experiment in which to drive progress in the domain toward eventual impact, but we think it is the right direction for the community now. If this task is too ambitious, another crucial project would be to further develop the semantics of neural network libraries and underlying computation primitives, to give frameworks like ONNX, Pytorch, CUDA, etc., formal semantics beyond what exists now in VNN-LIB, which would be a step toward addressing several existing challenges in soundness, usage of neural networks in broader systems, and that neural networks are not really just mathematical functions.

Acknowledgments

The author is grateful to useful conversations with many in the community and at Allerton 2025, and apologies if any relevant papers were missed. The material presented in this paper is based upon work supported by the National Science Foundation (NSF) through grant numbers 2220401 and 2220426 and the Defense Advanced Research Projects Agency (DARPA) under contract number FA8750-23-C-0518. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA or NSF.

References

[1] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient smt solver for verifying deep neural networks," in *Computer Aided Verification (CAV'17)*, R. Majumdar and V. Kunčak, Eds. Cham: Springer International Publishing, 2017, pp. 97–117.

[2] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, "Safety verification of deep neural networks," in *Computer Aided Verification*, R. Majumdar and V. Kunčak, Eds. Cham: Springer International Publishing, 2017, pp. 3–29.

[3] L. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, L. Daniel, D. Boning, and I. Dhillon, "Towards fast computation of certified robustness for ReLU networks," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 7 2018, pp. 5276–5285.

[4] S. Dutta, S. Jha, S. Sankaranarayanan, and A. Tiwari, "Output range analysis for deep feedforward neural networks," in *NASA Formal Methods*, A. Dutle, C. Muñoz, and A. Narkawicz, Eds. Cham: Springer International Publishing, 2018, pp. 121–138.

[5] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Formal security analysis of neural networks using symbolic intervals," in *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, Aug. 2018, pp. 1599–1614.

[6] ——, "Efficient formal safety analysis of neural networks," in *NeurIPS*, 2018, pp. 6369–6379.

[7] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev, "Ai2: Safety and robustness certification of neural networks with abstract interpretation," in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 3–18.

[8] W. Xiang, H.-D. Tran, and T. T. Johnson, "Output reachable set estimation and verification for multilayer neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5777–5783, 2018.

[9] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[10] R. R. Bunel, I. Turkaslan, P. Torr, P. Kohli, and P. K. Mudigonda, "A unified view of piecewise linear neural network verification," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[11] V. Tjeng, K. Y. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," in *International Conference on Learning Representations*, 2019.

[12] G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. Vechev, "Fast and effective robustness certification," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 10 802–10 813.

[13] G. Singh, T. Gehr, M. Püschel, and M. Vechev, "An abstract domain for certifying neural networks," *Proc. ACM Program. Lang.*, vol. 3, no. POPL, Jan. 2019.

[14] P. Prabhakar and Z. Rahimi Afzal, "Abstraction based output range analysis for neural networks," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[15] G. Katz, D. A. Huang, D. Ibeling, K. Julian, C. Lazarus, R. Lim, P. Shah, S. Thakoor, H. Wu, A. Zeljić, D. L. Dill, M. J. Kochenderfer, and C. Barrett, "The marabou framework for verification and analysis of deep neural networks," in *Computer Aided Verification (CAV'19)*,

I. Dillig and S. Tasiran, Eds. Springer International Publishing, 2019, pp. 443–452.

[16] H.-D. Tran, D. Manzanas Lopez, P. Musau, X. Yang, L. V. Nguyen, W. Xiang, and T. T. Johnson, "Star-based reachability analysis of deep neural networks," in *Formal Methods – The Next 30 Years*, M. H. ter Beek, A. McIver, and J. N. Oliveira, Eds. Cham: Springer International Publishing, 2019, pp. 670–686.

[17] H.-D. Tran, S. Bak, W. Xiang, and T. T. Johnson, "Verification of deep convolutional neural networks using imagestars," in *Computer Aided Verification*, S. K. Lahiri and C. Wang, Eds. Cham: Springer International Publishing, 2020, pp. 18–42.

[18] C. Urban, M. Christakis, V. Wüstholz, and F. Zhang, "Perfectly parallel fairness certification of neural networks," *Proc. ACM Program. Lang.*, vol. 4, no. OOPSLA, Nov. 2020.

[19] H.-D. Tran, X. Yang, D. Manzanas Lopez, P. Musau, L. V. Nguyen, W. Xiang, S. Bak, and T. T. Johnson, "Nnv: The neural network verification tool for deep neural networks and learning-enabled cyber-physical systems," in *Computer Aided Verification*, S. K. Lahiri and C. Wang, Eds. Cham: Springer International Publishing, 2020, pp. 3–17.

[20] W. Xiang, H. Tran, X. Yang, and T. T. Johnson, "Reachable set estimation for neural network control systems: A simulation-guided approach," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, pp. 1–10, 2020.

[21] H. Zhang, H. Chen, C. Xiao, S. Gowal, R. Stanforth, B. Li, D. Boning, and C.-J. Hsieh, "Towards stable and efficient training of verifiably robust neural networks," in *International Conference on Learning Representations*, 2020.

[22] S. Bak, C. Liu, and T. T. Johnson, "The second international verification of neural networks competition (VNN-COMP 2021): Summary and results," *CoRR*, vol. abs/2109.00498, 2021. [Online]. Available: https://arxiv.org/abs/2109.00498

[23] M. N. Müller, G. Makarchuk, G. Singh, M. Püschel, and M. Vechev, "Prima: general and precise neural network certification via scalable convex hull approximations," *Proc. ACM Program. Lang.*, vol. 6, no. POPL, Jan. 2022.

[24] M. N. Müller, C. Brix, S. Bak, C. Liu, and T. T. Johnson, "The third international verification of neural networks competition (vnn-comp 2022): Summary and results," 2022. [Online]. Available: https://arxiv.org/abs/2212.10376

[25] P. Prabhakar, "Bisimulations for neural network reduction," in *Verification, Model Checking, and Abstract Interpretation*, B. Finkbeiner and T. Wies, Eds. Cham: Springer International Publishing, 2022, pp. 285–300.

[26] C. Brix, M. N. Müller, S. Bak, T. T. Johnson, and C. Liu, "First three years of the international verification of neural networks competition (vnn-comp)," *Int. J. Softw. Tools Technol. Transf.*, vol. 25, no. 3, p. 329–339, May 2023. [Online]. Available: https://doi.org/10.1007/s10009-023-00703-4

[27] C. Brix, S. Bak, C. Liu, and T. T. Johnson, "The fourth international verification of neural networks competition (vnn-comp 2023): Summary and results," 2023. [Online]. Available: https://arxiv.org/abs/2312.16760

[28] S. Munakata, C. Urban, H. Yokoyama, K. Yamamoto, and K. Munakata, "Verifying attention robustness of deep neural networks against semantic perturbations," in *NASA Formal Methods*, K. Y. Rozier and S. Chaudhuri, Eds. Cham: Springer Nature Switzerland, 2023, pp. 37–61.

[29] S. Ugare, D. Banerjee, S. Misailovic, and G. Singh, "Incremental verification of neural networks," *Proc. ACM Program. Lang.*, vol. 7, no. PLDI, Jun. 2023.

[30] T. Ladner and M. Althoff, "Automatic abstraction refinement in neural network verification using sensitivity analysis," in *Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '23. New York, NY, USA: Association for Computing Machinery, 2023.

[31] ——, "Exponent relaxation of polynomial zonotopes and its applications in formal neural network verification," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 19, pp. 21 304–21 311, Mar. 2024. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/30125

[32] P. Habeeb and P. Prabhakar, "Approximate conformance verification of deep neural networks," in *NASA Formal Methods*, N. Benz, D. Gopinath, and N. Shi, Eds. Cham: Springer Nature Switzerland, 2024, pp. 223–238.

[33] C. Brix, S. Bak, T. T. Johnson, and H. Wu, "The fifth international verification of neural networks competition (vnn-comp 2024): Summary and results," 2024. [Online]. Available: https://arxiv.org/abs/2412.19985

[34] A. M. Tumlin, D. Manzanas Lopez, P. Robinette, Y. Zhao, T. Derr, and T. T. Johnson, "Fairnnv: The neural network verification tool for certifying fairness," in *Proceedings of the 5th ACM International Conference on AI in Finance*, ser. ICAIF '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 36–44.

[35] A. Kabaha and D. D. Cohen, "Verification of neural networks' global robustness," *Proc. ACM Program. Lang.*, vol. 8, no. OOPSLA, Apr. 2024.

[36] D. Zhou, C. Brix, G. A. Hanasusanto, and H. Zhang, "Scalable neural network verification with branch-and-bound inferred cutting planes," in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Curran Associates, Inc., 2024, pp. 29 324–29 353.

[37] Z. Shi, Q. Jin, Z. Kolter, S. Jana, C.-J. Hsieh, and H. Zhang, "Neural network verification with branch-and-bound for general nonlinearities," in *Tools and Algorithms for the Construction and Analysis of Systems*, A. Gurfinkel and M. Heule, Eds. Cham: Springer Nature Switzerland, 2025, pp. 315–335.

[38] J. Mour and D. Drachsler-Cohen, "Robustness verification of multi-label neural network classifiers," in *Static Analysis*, R. Giacobazzi and A. Gorla, Eds. Cham: Springer Nature Switzerland, 2025, pp. 327–351.

[39] T. Ladner, M. Eichelbeck, and M. Althoff, "Formal verification of graph convolutional networks with uncertain node features and uncertain graph structure," *Transactions on Machine Learning Research*, 2025.

[40] S. Sasaki, D. M. Lopez, P. K. Robinette, and T. T. Johnson, "Robustness verification of video classification neural networks," in *2025 IEEE/ACM 13th International Conference on Formal Methods in Software Engineering (FormaliSE)*, 2025, pp. 22–33.

[41] H.-D. Tran, S. W. Choi, Y. Li, Q. Liu, H. Okamoto, B. Hoxha, and G. Fainekos, "Starv: A qualitative and quantitative verification tool for learning-enabled systems," in *Computer Aided Verification*, R. Piskac and Z. Rakamarić, Eds. Cham: Springer Nature Switzerland, 2025, pp. 376–394.

[42] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C.-J. Hsieh, and Z. Kolter, "Beta-crown: efficient bound propagation with per-neuron split constraints for neural network robustness verification," in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NIPS '21. Red Hook, NY, USA: Curran Associates Inc., 2021.

[43] D. Zhou, C. Brix, G. A. Hanasusanto, and H. Zhang, "Scalable neural network verification with branch-and-bound inferred cutting planes," in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Curran Associates, Inc., 2024, pp. 29 324–29 353.

[44] V. Nagisetty, L. Graves, G. Pan, P. Jha, and V. Ganesh, "CGDTest: A constrained gradient descent algorithm for testing neural networks," 2023. [Online]. Available: https://arxiv.org/abs/2304.01826

[45] C. Brix and T. Noll, "Debona: Decoupled boundary network analysis for tighter bounds and faster adversarial robustness proofs," *CoRR*, vol. abs/2006.09040, 2020.

[46] F. Toledo, D. Shriver, S. Elbaum, and M. B. Dwyer, "Distribution models for falsification and verification of dnns," in *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '21. IEEE Press, 2022, p. 317–329.

[47] D. Shriver, S. Elbaum, and M. B. Dwyer, "Dnnv: A framework for deep neural network verification," in *Computer Aided Verification*, A. Silva and K. R. M. Leino, Eds. Cham: Springer International Publishing, 2021, pp. 137–150.

[48] J. Ferlez, H. Khedr, and Y. Shoukry, "Fast BATLLNN: Fast box analysis of two-level lattice neural networks," in *Proceedings of the 25th ACM International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3501710.3519533

[49] C. Ferrari, M. N. Mueller, N. Jovanović, and M. Vechev, "Complete verification via multi-neuron relaxation guided branch-and-bound," in *International Conference on Learning Representations*, 2022.

[50] S. Demarchi, D. Guidotti, L. Pulina, and A. Tacchella, "Never2: learning and verification of neural networks," *Soft Computing*, vol. 28, no. 19, pp. 11 647–11 665, 2024.

[51] H. Duong, T. Nguyen, and M. B. Dwyer, "NeuralSAT: A high-performance verification tool for deep neural networks," in *Computer Aided Verification*, R. Piskac and Z. Rakamarić, Eds. Cham: Springer Nature Switzerland, 2025, pp. 409–423.

[52] S. Bak, "nnenum: Verification of relu neural networks with optimized abstraction refinement," in *NASA Formal Methods*, A. Dutle, M. M. Moscato, L. Titolo, C. A. Muñoz, and I. Perez, Eds. Cham: Springer International Publishing, 2021, pp. 19–36.

[53] D. M. Lopez, S. W. Choi, H.-D. Tran, and T. T. Johnson, "NNV 2.0: The neural network verification tool," in *Computer Aided Verification*, C. Enea and A. Lal, Eds. Cham: Springer Nature Switzerland, 2023, pp. 397–412.

[54] T. Wei, H. Hu, L. Marzari, K. S. Yun, P. Niu, X. Luo, and C. Liu, "ModelVerification.jl: A comprehensive toolbox for formally verifying deep neural networks," in *Computer Aided Verification*, R. Piskac and Z. Rakamarić, Eds. Cham: Springer Nature Switzerland, 2025, pp. 395–408.

[55] R. Bunel, J. Lu, I. Turkaslan, P. H. Torr, P. Kohli, and M. P. Kumar, "Branch and bound for piecewise linear neural network verification," *Journal of Machine Learning Research*, vol. 21, no. 42, pp. 1–39, 2020. [Online]. Available: http://jmlr.org/papers/v21/19-468.html

[56] H. Khedr, J. Ferlez, and Y. Shoukry, "Peregrinn: Penalized-relaxation greedy neural network verifier," in *Computer Aided Verification*, A. Silva and K. R. M. Leino, Eds. Cham: Springer International Publishing, 2021, pp. 287–300.

[57] A. Lemesle, J. Lehmann, and T. L. Gall, "Neural network verification with pyrat," 2024. [Online]. Available: https://arxiv.org/abs/2410.23903

[58] J. A. Vincent and M. Schwager, "Reachable polyhedral marching (rpm): A safety verification algorithm for robotic systems with deep neural network components," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9029–9035.

[59] J. Smith, J. Allen, V. Swaminathan, and Z. Zhang, "Refutation-based adversarial robustness verification of deep neural networks," in *In Formal Methods for ML-Enabled Autonomous Systems*, 2021.

[60] E. Botoeva, P. Kouvaros, J. Kronqvist, A. Lomuscio, and R. Misener, "Efficient verification of relu-based neural networks via dependency analysis," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 3291–3299, Apr. 2020.

[61] P. Henriksen and A. Lomuscio, "Efficient neural network verification via adaptive refinement and adversarial search," in *European Conference on Artificial Intelligence (ECAI)*. IOS Press, 2020, pp. 2513–2520.

[62] D. Gopinath, C. Pasareanu, and B. Kadron, "Analysis of taxinet neural network," in *NASA-Boeing Autonomy V&V Project: NASA V&V Tool Demonstration*, 2021.

[63] D. Manzanas Lopez and T. T. Johnson, "Empirical analysis of benchmark generation for the verification of neural network image classifiers," in *Bridging the Gap Between AI and Reality*, B. Steffen, Ed. Cham: Springer Nature Switzerland, 2024, pp. 331–347.

[64] M. P. Owen, A. Panken, R. Moss, L. Alvarez, and C. Leeper, "Acas xu: Integrated collision avoidance and detect and avoid capability for uas," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 2019, pp. 1–10.

[65] S. Bak and H.-D. Tran, "Neural network compression of acas xu early prototype is unsafe: Closed-loop verification through quantized state backreachability," in *NASA Formal Methods*, J. V. Deshmukh, K. Havelund, and I. Perez, Eds. Cham: Springer International Publishing, 2022, pp. 280–298.

[66] D. Kirov, S. F. Rollini, L. Di Guglielmo, and D. Cofer, "Formal verification of a neural network based prognostics system for aircraft equipment," in *Bridging the Gap Between AI and Reality*, B. Steffen, Ed. Cham: Springer Nature Switzerland, 2024, pp. 225–240.

[67] D. Kirov, S. F. Rollini, R. Chandrahas, S. R. Chandupatla, and R. Sawant, "Benchmark: Object detection for maritime search and rescue," in *Bridging the Gap Between AI and Reality*, B. Steffen, Ed. Cham: Springer Nature Switzerland, 2024, pp. 305–310.

[68] P. K. Robinette, D. M. Lopez, and T. T. Johnson, "Benchmark: Neural network malware classification," in *Bridging the Gap Between AI and Reality*, B. Steffen, Ed. Cham: Springer Nature Switzerland, 2024, pp. 291–298.

[69] D. M. Lopez, P. Musau, H.-D. Tran, S. Dutta, T. J. Carpenter, R. Ivanov, and T. T. Johnson, "Arch-comp19 category report: Artificial intelligence and neural network control systems (ainncs) for continuous and hybrid systems plants," in *ARCH19. 6th International Workshop on Applied Verification of Continuous and Hybrid Systems*, ser. EPiC Series in Computing, G. Frehse and M. Althoff, Eds., vol. 61. EasyChair, 2019, pp. 103–119.

[70] T. T. Johnson, D. M. Lopez, P. Musau, H.-D. Tran, E. Botoeva, F. Leofante, A. Maleki, C. Sidrane, J. Fan, and C. Huang, "Arch-comp20 category report: Artificial intelligence and neural network control systems (ainncs) for continuous and hybrid systems plants," in *ARCH20. 7th International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH20)*, ser. EPiC Series in Computing, G. Frehse and M. Althoff, Eds., vol. 74. EasyChair, 2020, pp. 107–139.

[71] T. T. Johnson, D. M. Lopez, L. Benet, M. Forets, S. Guadalupe, C. Schilling, R. Ivanov, T. J. Carpenter, J. Weimer, and I. Lee, "Arch-comp21 category report: Artificial intelligence and neural network control systems (ainncs) for continuous and hybrid systems plants," in *8th International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH21)*, ser. EPiC Series in Computing, G. Frehse and M. Althoff, Eds., vol. 80. EasyChair, 2021, pp. 90–119.

[72] D. M. Lopez, M. Althoff, L. Benet, X. Chen, J. Fan, M. Forets, C. Huang, T. T. Johnson, T. Ladner, W. Li, C. Schilling, and Q. Zhu, "Arch-comp22 category report: Artificial intelligence and neural network control systems (ainncs) for continuous and hybrid systems plants," in *Proceedings of 9th International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH22)*, ser. EPiC Series in Computing, G. Frehse, M. Althoff, E. Schoitsch, and J. Guiochet, Eds., vol. 90. EasyChair, 2022, pp. 142–184.

[73] D. M. Lopez, M. Althoff, M. Forets, T. T. Johnson, T. Ladner, and C. Schilling, "Arch-comp23 category report: Artificial intelligence and neural network control systems (ainncs) for continuous and hybrid systems plants," in *Proceedings of 10th International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH23)*, ser. EPiC Series in Computing, G. Frehse and M. Althoff, Eds., vol. 96. EasyChair, 2023, pp. 89–125.

[74] D. M. Lopez, M. Althoff, L. Benet, C. Blab, M. Forets, Y. Jia, T. T. Johnson, M. Kranzl, T. Ladner, L. Linauer, P. Neubauer, S. Neubauer, C. Schilling, H. Zhang, and X. Zhong, "Arch-comp24 category report: Artificial intelligence and neural network control systems (ainncs) for continuous and hybrid systems plants," in *Proceedings of the 11th Int. Workshop on Applied Verification for Continuous and Hybrid Systems*, ser. EPiC Series in Computing, G. Frehse and M. Althoff, Eds., vol. 103. EasyChair, 2024, pp. 64–121. [Online]. Available: /publications/paper/WsgX

[75] M. Althoff, "An introduction to CORA 2015," in *Workshop on Applied Verification for Continuous and Hybrid Systems*, 2015, p. 120–151.

[76] N. Kochdumper, C. Schilling, M. Althoff, and S. Bak, "Open- and closed-loop neural network verification using polynomial zonotopes," in *NASA Formal Methods*. Springer, 2023, pp. 16–36.

[77] A. Harapanahalli, S. Jafarpour, and S. Coogan, "immrax: A parallelizable and differentiable toolbox for interval analysis and mixed monotone reachability in jax," *IFAC-PapersOnLine*, vol. 58, no. 11, pp. 75–80, 2024, 8th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896324005275

[78] A. Harapanahalli and S. Coogan, "Certified robust invariant polytope training in neural controlled odes," 2024.

[79] S. Jafarpour, A. Harapanahalli, and S. Coogan, "Efficient interaction-aware interval analysis of neural network feedback loops," *IEEE Transactions on Automatic Control*, vol. 69, no. 12, pp. 8706–8721, 2024.

[80] S. Bogomolov, M. Forets, G. Frehse, K. Potomkin, and C. Schilling, "JuliaReach: A toolbox for set-based reachability," in *Proc. of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, 2019, p. 39–44.

[81] S. Dutta, S. Jha, S. Sankaranarayanan, and A. Tiwari, "Learning and verification of feedback control systems using feedforward neural networks," *IFAC-PapersOnLine*, vol. 51, no. 16, pp. 151 – 156, 2018, iFAC Conference on Analysis and Design of Hybrid Systems ADHS 2018.

[82] S. Dutta, X. Chen, and S. Sankaranarayanan, "Reachability analysis for neural feedback systems using regressive polynomial rule inference," in

ACM International Conference on Hybrid Systems: Computation and Control, HSCC, 2019, pp. 157–168.

[83] C. Huang, J. Fan, X. Chen, W. Li, and Q. Zhu, "POLAR: A polynomial arithmetic framework for verifying neural-network controlled systems," in *International Symposium on Automated Technology for Verification and Analysis (ATVA)*, 2022.

[84] C. Sidrane and M. J. Kochenderfer, "OVERT: Verification of nonlinear dynamical systems with neural network controllers via overapproximation," *Safe Machine Learning workshop at ICLR*, 2019.

[85] C. Huang, J. Fan, W. Li, X. Chen, and Q. Zhu, "Reachnn: Reachability analysis of neural-network controlled systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 5s, pp. 1–22, 2019.

[86] J. Fan, C. Huang, W. Li, X. Chen, and Q. Zhu, "Reachnn*: A tool for reachability analysis ofneural-network controlled systems," in *International Symposium on Automated Technology for Verification and Analysis (ATVA)*, 2020.

[87] M. E. Akintunde, E. Botoeva, P. Kouvaros, and A. Lomuscio, "Formal verification of neural agents in non-deterministic environments," in *International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2020, pp. 25–33.

[88] R. Ivanov, J. Weimer, R. Alur, G. J. Pappas, and I. Lee, "Verisig: Verifying safety properties of hybrid systems with neural network controllers," in *International Conference on Hybrid Systems: Computation and Control*, ser. HSCC. ACM, 2019, p. 169–178.

[89] R. Ivanov, T. Carpenter, J. Weimer, R. Alur, G. J. Pappas, and I. Lee, "Verisig 2.0: Verification of neural network controllers using taylor model preconditioning," in *International Conference on Computer-Aided Verification*, 2021.

[90] H.-D. Tran, W. Xiang, and T. T. Johnson, "Verification approaches for learning-enabled autonomous cyber–physical systems," *IEEE Design and Test*, vol. 39, no. 1, pp. 24–34, 2022.

[91] A. Albarghouthi *et al.*, "Introduction to neural network verification," *Foundations and Trends® in Programming Languages*, vol. 7, no. 1–2, pp. 1–157, 2021.

[92] C. Urban and A. Miné, "A review of formal methods applied to machine learning," 2021. [Online]. Available: https://arxiv.org/abs/2104.02466

[93] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, and M. J. Kochenderfer, *Algorithms for Verifying Deep Neural Networks*. Now Foundations and Trends, 2021.

[94] S. A. Seshia, D. Sadigh, and S. S. Sastry, "Toward verified artificial intelligence," *Commun. ACM*, vol. 65, no. 7, p. 46–55, Jun. 2022.

[95] H.-D. Tran, F. Cai, M. L. Diego, P. Musau, T. T. Johnson, and X. Koutsoukos, "Safety verification of cyber-physical systems with reinforcement learning control," *ACM Trans. Embed. Comput. Syst.*, vol. 18, no. 5s, oct 2019.

[96] Z. Li, J. Huang, and M. Naik, "Scallop: A language for neurosymbolic programming," *Proc. ACM Program. Lang.*, vol. 7, no. PLDI, Jun. 2023.

[97] S. S. Serbinowska, P. Robinette, G. Karsai, and T. T. Johnson, "Formalizing stateful behavior trees," in Proceedings Sixth International Workshop on *Formal Methods for Autonomous Systems,* Manchester, UK, 11th and 12th of November 2024, ser. Electronic Proceedings in Theoretical Computer Science, M. Luckcuck and M. Xu, Eds., vol. 411. Open Publishing Association, 2024, pp. 201–218.

[98] S. S. Serbinowska, D. Manzanas Lopez, D. T. Nguyen, and T. T. Johnson, "Neuro-symbolic behavior trees (nsbts) and their verification," in *Proceedings of the International Conference on Neuro-symbolic Systems*, ser. Proceedings of Machine Learning Research, G. Pappas, P. Ravikumar, and S. A. Seshia, Eds., vol. 288. PMLR, 28–30 May 2025, pp. 409–423. [Online]. Available: https://proceedings.mlr.press/v288/serbinowska25a.html

[99] S. Sasaki, D. Manzanas Lopez, and T. T. Johnson, "Neurosymbolic finite and pushdown automata: Improved multimodal reasoning versus vision language models (vlms)," in *Proceedings of the International Conference on Neuro-symbolic Systems*, ser. Proceedings of Machine Learning Research, G. Pappas, P. Ravikumar, and S. A. Seshia, Eds., vol. 288. PMLR, 28–30 May 2025, pp. 170–187. [Online]. Available: https://proceedings.mlr.press/v288/sasaki25a.html

[100] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. De Raedt, "Deepproblog: Neural probabilistic logic programming," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[101] S. Chaudhuri, K. Ellis, O. Polozov, R. Singh, A. Solar-Lezama, and Y. Yue, "Neurosymbolic programming," *Foundations and Trends in Programming Languages*, vol. 7, no. 3, pp. 158–243, 2021.

[102] L. C. Cordeiro, M. L. Daggitt, J. Girard-Satabin, O. Isac, T. T. Johnson, G. Katz, E. Komendantskaya, A. Lemesle, E. Manino, A. Šinkarovs, and H. Wu, "Neural network verification is a programming language challenge," in *Programming Languages and Systems*, V. Vafeiadis, Ed. Cham: Springer Nature Switzerland, 2025, pp. 206–235.

[103] D. Zombori, B. Bánhelyi, T. Csendes, I. Megyeri, and M. Jelasity, "Fooling a complete neural network verifier," in *International Conference on Learning Representations*, 2021.

[104] K. Jia and M. Rinard, "Exploiting verified neural networks via floating point numerical error," in *Static Analysis*, C. Drăgoi, S. Mukherjee, and K. Namjoshi, Eds. Cham: Springer International Publishing, 2021, pp. 191–205.

[105] R. Elsaleh and G. Katz, "Delbugv: Delta-debugging neural network verifiers," in *2023 Formal Methods in Computer-Aided Design (FMCAD)*, 2023, pp. 34–43.

[106] X. Zhou, H. Xu, A. Xu, Z. Shi, C.-J. Hsieh, and H. Zhang, "Testing neural network verifiers: A soundness benchmark with hidden counterexamples," *CoRR*, vol. abs/2412.03154, 2024.

[107] M. König, A. W. Bosman, H. H. Hoos, and J. N. van Rijn, "Critically assessing the state of the art in neural network verification," *Journal of Machine Learning Research*, vol. 25, no. 12, pp. 1–53, 2024.

[108] A. Szász, B. Bánhelyi, and M. Jelasity, "No soundness in the real world: On the challenges of the verification of deployed neural networks," in *Forty-second International Conference on Machine Learning*, 2025.

[109] S. Demarchi, D. Guidotti, L. Pulina, and A. Tacchella, "Supporting standardization of neural networks verification with vnnlib and coconet," in *Proceedings of the 6th Workshop on Formal Methods for ML-Enabled Autonomous Systems*, ser. Kalpa Publications in Computing, N. Narodytska, G. Amir, G. Katz, and O. Isac, Eds., vol. 16. EasyChair, 2023, pp. 47–58.

[110] M. Sotoudeh and A. V. Thakur, "Provable repair of deep neural networks," in *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, ser. PLDI 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 588–603.

[111] M. Usman, D. Gopinath, Y. Sun, Y. Noller, and C. S. Păsăreanu, "Nnrepair: Constraint-based repair of neural network classifiers," in *Computer Aided Verification*, A. Silva and K. R. M. Leino, Eds. Cham: Springer International Publishing, 2021, pp. 3–25.

[112] X. Yang, T. Yamaguchi, H.-D. Tran, B. Hoxha, T. T. Johnson, and D. Prokhorov, "Neural network repair with reachability analysis," in *Formal Modeling and Analysis of Timed Systems*, S. Bogomolov and D. Parker, Eds. Cham: Springer International Publishing, 2022, pp. 221–236.

[113] G. Dong, J. Sun, X. Wang, X. Wang, and T. Dai, "Towards repairing neural networks correctly," in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, 2021, pp. 714–725.

[114] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023.

[115] D. Groeneveld, I. Beltagy, P. Walsh, A. Bhagia, R. Kinney, O. Tafjord, A. H. Jha, H. Ivison, I. Magnusson, Y. Wang, S. Arora, D. Atkinson, R. Authur, K. R. Chandu, A. Cohan, J. Dumas, Y. Elazar, Y. Gu, J. Hessel, T. Khot, W. Merrill, J. Morrison, N. Muennighoff, A. Naik, C. Nam, M. E. Peters, V. Pyatkin, A. Ravichander, D. Schwenk, S. Shah, W. Smith, E. Strubell, N. Subramani, M. Wortsman, P. Dasigi, N. Lambert, K. Richardson, L. Zettlemoyer, J. Dodge, K. Lo, L. Soldaini, N. A. Smith, and H. Hajishirzi, "Olmo: Accelerating the science of language models," 2024. [Online]. Available: https://arxiv.org/abs/2402.00838

[116] T. OLMo, P. Walsh, L. Soldaini, D. Groeneveld, K. Lo, S. Arora, A. Bhagia, Y. Gu, S. Huang, M. Jordan, N. Lambert, D. Schwenk, O. Tafjord, T. Anderson, D. Atkinson, F. Brahman, C. Clark, P. Dasigi, N. Dziri, M. Guerquin, H. Ivison, P. W. Koh, J. Liu, S. Malik, W. Merrill, L. J. V. Miranda, J. Morrison, T. Murray, C. Nam, V. Pyatkin, A. Rangapur, M. Schmitz, S. Skjonsberg, D. Wadden, C. Wilhelm, M. Wilson, L. Zettlemoyer, A. Farhadi, N. A. Smith, and H. Hajishirzi, "2 olmo 2 furious," 2025.

[117] M. Casadio, T. Dinkar, E. Komendantskaya, L. Arnaboldi, M. L. Daggitt, O. Isac, G. Katz, V. Rieser, and O. Lemon, "Nlp verification: towards a general methodology for certifying robustness," *European Journal of Applied Mathematics*, p. 1–58, 2025.

[118] M. L. Daggitt, W. Kokke, R. Atkey, E. Komendantskaya, N. Slusarz, and L. Arnaboldi, "Vehicle: Bridging the Embedding Gap in the Verification of Neuro-Symbolic Programs," in *10th International Conference on Formal Structures for Computation and Deduction (FSCD 2025)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), M. Fernández, Ed., vol. 337. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025, pp. 2:1–2:20.

[119] Z. Shi, H. Zhang, K.-W. Chang, M. Huang, and C.-J. Hsieh, "Robustness verification for transformers," in *International Conference on Learning Representations*, 2020.

[120] G. Bonaert, D. I. Dimitrov, M. Baader, and M. Vechev, "Fast and precise certification of transformers," in *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, ser. PLDI 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 466–481.

[121] B. H.-C. Liao, C.-H. Cheng, H. Esen, and A. Knoll, "Are transformers more robust? towards exact robustness verification for transformers," in *Computer Safety, Reliability, and Security*, J. Guiochet, S. Tonetta, and F. Bitsch, Eds. Cham: Springer Nature Switzerland, 2023, pp. 89–103.

[122] P. Belcak, G. Heinrich, S. Diao, Y. Fu, X. Dong, S. Muralidharan, Y. C. Lin, and P. Molchanov, "Small language models are the future of agentic ai," 2025.

[123] T. Hoare, J. Misra, G. T. Leavens, and N. Shankar, *The Verified Software Initiative: A Manifesto*, 1st ed. New York, NY, USA: Association for Computing Machinery, 2021, p. 81–92.

[124] T. Hoare, "The verifying compiler: A grand challenge for computing research," *J. ACM*, vol. 50, no. 1, p. 63–69, Jan. 2003.

[125] X. Leroy, "Formal verification of a realistic compiler," *Commun. ACM*, vol. 52, no. 7, p. 107–115, Jul. 2009.

[126] A. W. Appel, "Verified software toolchain," in *Programming Languages and Systems*, G. Barthe, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1–17.